Defining Tolerance: Impacts of Delay and Disruption when Managing Challenged Networks

Edward J. Birrane, III.[1]
Johns Hopkins University Applied Physics Laboratory, Laurel, Maryland,20723

Scott C. Burleigh[2]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109

and

Dr. Vint Cerf[3]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 91109

Challenged networks exhibit irregularities in their communication performance stemming from node mobility, power constraints, and impacts from the operating environment. These irregularities manifest as high signal propagation delay and frequent link disruption. Understanding those limits of link disruption and propagation delay beyond which core networking features fail is an ongoing area of research. Various wireless networking communities propose tools and techniques that address these phenomena. Emerging standardization activities within the Internet Research Task Force (IRTF) and the Consultative Committee for Space Data Systems (CCSDS) look to build upon both this experience and scalability analysis. Successful research in this area is predicated upon identifying enablers for common communication functions (notably node discovery, duplex communication, state caching, and link negotiation) and how increased disruptions and delays affect their feasibility within the network. Networks that make fewer assumptions relating to these enablers provide more universal service. Specifically, reliance on node discovery and link negotiation results in network-specific operational concepts rather than scalable technical solutions. Fundamental to this debate are the definitions, assumptions, operational concepts, and anticipated scaling of these networks. This paper presents the commonalities and differences between delay and disruption tolerance, including support protocols and critical enablers. We present where and how these tolerances differ. We propose a set of use cases that must be accommodated by any standardized delay-tolerant network and discuss the implication of these on existing tool development.

I. Introduction

The field of wireless network communications continues to develop as high-capability, low power processors, transceivers, energy harvesters, and mobility drivers evolve. The literature is well populated with applications of these technologies towards the construction of wireless networks, including wireless sensor networks, mobile ad-hoc networks, interplanetary internetworks, and surveillance networks.

These networks are generally referred to as “Challenged” networks (CN), as they operate without reliance on the power and communications infrastructure supporting wired and structured wireless networks. Typically challenged networks comprise wireless data transmission amongst mobile platforms operating within scalable environments hostile to radio frequency communication.

---

[1] Senior Professional Staff, Space Department, 11100 Johns Hopkins Rd/7-108, MB
[2] Principal Engineer, Systems Engineering, 4800 Oak Grove Drive, m/s 301-490.
[3] Distinguished Visiting Scientist.

The operating environment plays a significant role in the ultimate structure and capability of the CN. Networks that must span vast (interplanetary) distances incur large signal propagation delays relative to their data transmission rates. Smaller networks may still span a large enough geographic region to preclude dense node population, or even to guarantee a never-partitioned network. The links that comprise communications paths within CNs are shaped by a variety of environmental factors, including geometry, natural interference, man-made interference, node mobility, and antenna pointing,

The root cause of per-link failure effects falls into two categories – delay and disruption – both of which impose varying degrees of latency on the communication of data amongst nodes. In terrestrial wired networks, delay tends to be negligible and disruption a symptom of network error. CNs incur increasing delay as nodes increase their relative distance. They also incur regular disruption as antenna pointing, range, and power cycling alter contact potential within the network. Significantly, within these networks, disruption ceases to be a relatively rare error condition and becomes a part of the *nominal* operation of the network.

Delay tolerant networking[1] is an emerging subset of challenged networking that addresses the problems of significantly delayed and disrupted links. There is debate within the community as to whether *Delay-Tolerant* Networking and *Disruption-Tolerant* Networking address separate networking issues, or whether one set of protocols and operational concepts fully addresses the concerns of the other. A network link may have very high delay while experiencing no disruption. Similarly, an often-disrupted link may experience no significant delay during its uptime. Fundamental to this debate are the definitions, assumptions, operational concepts, and anticipated scaling of these networks.

This paper proposes a model for addressing delay and disruption in CNs with the goal of converging engineering and research in this area. We present definitions of delay, disruption, and tolerance both at the link layer and at the network layer. We list the network characteristics and associated enablers assumed available to these networks. Finally, we discuss the impact of network design and operational concepts on protocol behavior. We conclude that tolerance of data communication latency is the root problem; that, due to the upper limit on the speed of light in a given medium, the set of available mitigations for latency resulting from delay is a proper subset of the mitigations available for latency resulting from disruptions. Therefore, tolerance of delays encompasses tolerance of disruptions and Delay-Tolerant Networking is the appropriate encompassing context as efforts to mitigate disruption without those mechanisms that also mitigate delay result in networks that lack scalability in mobility, size, or data rate.

The remainder of this paper is structured as follows. Section II discusses the motivation for this work. Section III introduces relevant terminology and associated concepts necessary to frame the tolerance discussion. Section IV outlines how delayed and disrupted networks function using the terminology introduced in this paper and identifies a pedagogical topology. Section V argues that delay tolerance is a superset of disruption tolerance in networks. Section VI summarizes this work and proposed future work based on optimizing special networking cases.

II. Motivation

Emerging standardization activities within the Internet Research Task Force (IRTF) and the Consultative Committee for Space Data Systems (CCSDS) rely upon both existing network experience and scalability analysis. However, the design of protocols that function given levels of link disruption and delay unsupportable in current wireless networks is an active area of research. This paper is motivated by the desire to more precisely define the characteristics and effects of disruption and delay to better focus this research and quantify progress to date. We believe such convergence will result in better standards faster. Fundamental to this effort is describing why existing wireless protocols are neither delay nor disruption tolerant and agreeing on the characteristics of tolerant protocols.

Wireless communications are constrained by the Media Access Control (MAC) problem. This problem asks when it is appropriate for a transmitter to emit a message and, symmetrically, when is it appropriate for a receiver to be powered and accept such a message. A variety of solutions[2] are employed by modern wireless MAC protocols, including carrier sensing[3], schedule synchronization[4], timing pulses[5], out-of-band communication[6], and multiplexing[7]. Common to all of these approaches is the assumption that round-trip communication amongst nodes must occur prior to the establishment of an application-data communication session. Errors based on standard wireless impairments such as overload from hidden terminals and signal attenuation (shadowing, reflection, diffraction, refraction, and absorption) are overcome by both uni-cast and multi-cast retransmission. Many protocols treat this recovery as a phase separate from nominal operation[8].

The MAC problem is solved for wireless networks at the data-link layer of the OSI model[9]. Higher-level protocols that perform network operations (versus link operations) exist at the network and session layers. Significantly, these protocols implement the data forwarding/routing functions necessary to meaningfully migrate data through the network. Several higher-order protocols[10,11] carry forward the link-layer assumption of bi-directional communication, flooding, and recovery separate from nominal operation. These protocols attempt to synchronize on some macro-network state as part of nominal operations. To do this, they must synchronize and perform their function before the state of the network changes again. As the amount of delay and disruption experienced by the network increases, these network-layer protocols cannot perform their function, even when the underlying data-link-layer protocols continue to function.

A motivating example is to observe how a round-trip network-layer protocol fails when delay and disruption combine in such a way as to prevent bi-directional data exchange in a single communication session. Consider a network topology from the Solar System Internet[12] whose nodes are planetary orbiters separated by interplanetary distances. The motion of the orbiters around their respective planets, the planets around the sun, and the pointing of orbiter antennae imply that the links between orbiters are down more often than they are up – this is a high-disruption environment. Similarly, the distances between the orbiters are significant -- Earth and Mars vary between 4.5 and 20.8 light-minutes apart at any given time. If the link between orbiters cannot be maintained long enough to support at least one round-trip communication (9 – 41 minutes) then any protocol that requires any round-trip communication to establish an application-data exchange will fail. This situation is illustrated in Fig. 1.

Space networks spanning interplanetary distances represent the most extreme case of link delay. However, there are several terrestrial wireless networks for which delays are smaller but which experience increased disruption based on more frequent mobility, more directional transmission, or more frequent link impairments. These networks suffer the same problem as space networks: communication sessions do not last long enough to support the bi-directional data exchange required by network-layer protocols to perform their functions. Similarly, high-speed wired networks used to close mission critical control loops do not tolerate protocol overhead and round-trip negotiation, despite having transmission latencies in the microseconds. The problems of delay and disruption are found in networks of varying sizes and speeds.

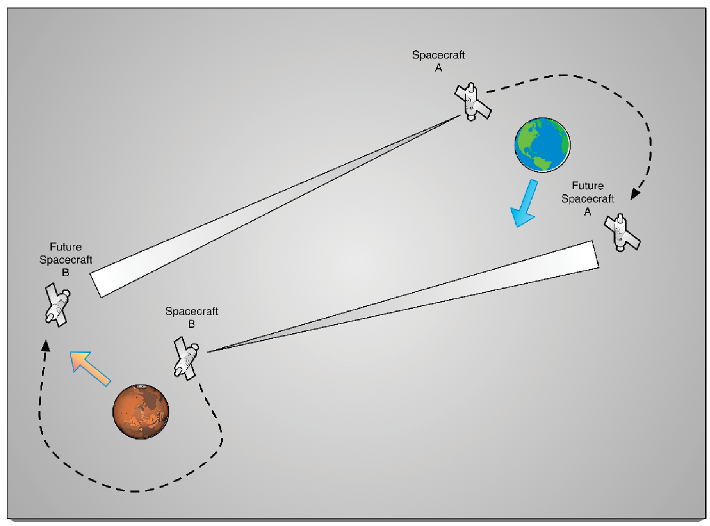

Figure 1. Celestial Motion Changes Network Geometry. *Directional transmission, planet rotation, orbits, and spacecraft pointing create a highly dynamic network topology.*

There is an open question in the research community as to whether protocol tolerance stems from tolerating increasing delays, increasing disruptions, or both. We claim protocols that operate in the high-delay, high-disruption space domain also address the needs of terrestrial, challenged networks. Certainly at the data-link layer both propagation delay and non-permanent disruption resulting in re-transmission resolve to the far side of the link waiting for its data. Any protocol that provides a model for this waiting (by either modeling delay or probabilistic number of retries) is tolerating the overall transmission delay and, thus, delay is a superset of disruption. However, the analysis becomes more complex when considering the entire network, where individual link problems may be overcome by alternate routing choices, packetization, and persistent storage at network nodes. Several approaches to disruption tolerance in wireless networking protocols either ignore disruption altogether and rely on higher-layer protocols (such as routing protocols) to handle data loss, or treat disruption as a transient error condition with a special protocol phase devoted to recovery. Recovery is different from tolerance. First, there is no recovery from a high propagation delay. Second, recovery from long-term disruption may require rediscovery of cached network state. Such resynchronization typically requires a large amount of activity on the network, such as broadcasts and packet floods. When the CN is delayed/disrupted so often as to be the rule rather than the exception these protocols spend most of their time in the recovery phase and do not transmit data. Clarifying this for the CN research community should result in a clearer approach to protocol development.

III. Definitions

Part of the confusion associated with delay/disruption tolerance stems from a lack of common definition of key terms. We present a set of relevant system-level definitions applicable to the challenged network space and critical to the understanding of delay and disruption tolerant networks. Understanding what is unique in the definitions of delay and disruption is predicated upon both a "bottoms-up" analysis of link states and a top-down analysis of network services.

A. Link Definitions

Links are the atomic unit of communication within computer networks. It follows that the relative capabilities and degradations of these links will have some aggregate affect on the network-level services that flow through them. In wireless networks a link is established amongst all nodes that share a given transmission footprint – wireless transmission is inherently multi-cast at the link level. While we focus on network-layer links between two nodes in a network, these definitions apply equally to the exchange of data amongst a larger set of receivers.

1) *Transmission Latency*: Typically a function of link data rate and propagation delay measured by the speed of light through the chosen medium. For the purpose of this discussion, we conflate these components into a single measure of the latency between placing data on a transmission queue at one node and retrieving it from a receiving queue at another node.

2) *Link Disruption*: the loss of communication across a data link. Disruption may be planned or unplanned and is considered a temporary impairment across the link. Permanent disruption would be considered a re-engineering of the network and not a link state. During the period of disruption, no information is successfully communicated across the link. Periods of minor link degradation, such as recoverable increases in bit error rate, are not considered disruption if data exchange continues to be possible across the link.

3) *Action Latency Limit*: the system-specific performance requirement associated with a maximum latency permitted between the occurrence of a network event and the application of an appropriate response to that event. Data transmitted within a network satisfies either an explicit or implicit action latency limit. For example, a communication network may need to fall back to a pre-provisioned recovery circuit within 500 milliseconds of loss of real-time data transfer across a main communications circuit. In this event, 500 milliseconds is the action latency of the network, which must detect the error and switch to the recovery circuit within this limit.

4) *High-Latency / Low-Latency Link*: A high-latency link is one in which the transmission latencies associated with round-trip communications exceed the action latency limit for either the specific link or for those paths utilizing the link[13]. A low-latency link supports round-trip communications within a given action latency limit. The definition is driven by the action latency limit and the volume of data that must be communicated as part of the action. For example, given a 100ms failover action latency in a real-time communications network, a link latency of 60ms results in a high-latency link because the round-trip exchange necessary to detect error results in a delay of 120ms, which is greater than the 100ms action time. Significantly, high-latency and low-latency links are not defined as a function of absolute time – a 120ms delay may result in a high-latency link. Conversely, a 1000ms lunar link may be considered low-latency if there is no driving performance metric.

B. Network Definitions

Multi-hop, multi-path networks support more complex operations than simply the sum of a series of individual links. Here we identify those terms that distinguish network-wide concepts from per-link performance.

1) *Network/Networking Functions*: The core technical services provided by modern computer networks, to include routing/forwarding, security, and management. Networks are those clusters of directly or indirectly connected computers that support these networking functions.

2) *Common Networking Enablers*: Those techniques that are used to support core functions in most modern packetized networks. In the context of this discussion, we identify the following enablers: node discovery, link negotiation, state caching, and bi-directional communication.

3) *Challenged Network*: A network unable to perform the aforementioned networking functions by relying solely on the aforementioned common networking enablers.

4) *Disrupted Network*: Disrupted networks are those that do not maintain a stable topology as part of their normal operating state. All networks experience downtime as part of dealing with node loss and other errors. However, disrupted networks always operate in this mode – the network is always (multi-) partitioned. Protocols that treat topological change as a special case or with a separate protocol phase will never reach a steady state in these networks. While this definition is made without regard to the delay associated with the network it is implied that the definition of "stable topology over time" is driven by the delay associated with the links in the network.

5) *Network Service*: A service that produces an intended result in response to an understood request, both of which being communicated through the network. Communication incurs transmission latencies. The user of a service requests a result within its guiding action latency limit.

6) *Quality of Service (QoS)*: Metrics associated with a service, such as minimal effective data rate over time. Services that meet their QoS requirements are functioning services.

7) *Network Tolerance*: The ability to satisfy QoS contracts in the face of unfavorable impacts on the network as long as doing so remains physically possible.

8) *Delay/Disruption Tolerant Network*: A Delay-Tolerant network continues to satisfy QoS contracts as delay in the network fluctuates. A Disruption-Tolerant network continues to satisfy QoS contracts as disruption in the network fluctuates. In both cases, fluctuation includes the increase in phenomena up to, but not beyond, the point at which QoS becomes impossible to support. Tolerance may require achieving traditional networking functions using alternate networking enablers.

IV. Delayed/Disrupted Tolerant Model

The Solar System Internet (SSI) concept supports a range of delays and disrupted links, making it a useful model for this discussion. We have distilled the most interest networking scenarios from this concept into the least non-trivial topology presented in Fig. 2. In this figure, fixed ground assets such as mission operation centers (MOCs) and science operations centers (SOCs) represent wired, low-latency communication paths. Orbiters and relays communicate with MOCs over high-latency links periodically disrupted by celestial motion. Ground entry points such as the Deep Space Network (DSN) are elided for clarity. It is sufficient for this discussion to assume that these entry points ferry information directly to MOCs.

Landers on planetary surfaces exploit mobility and form both surface mesh networks and hierarchical networks connected through relaying orbiters. This network example is used through the rest of this paper to discuss the protocol and algorithms used to provide tolerance to these phenomena.

We enumerate several defining observations of this model which must be addressed by any proposed tolerant, network-level protocol. These observations, and their associated implications for protocol operating within the model, are listed in Table 1.

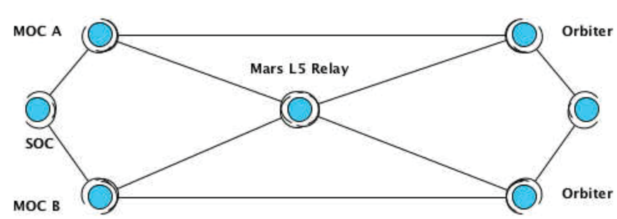

Figure 2. The topology of the SSI contains a representative mixture of delays and disruptions. *Low latency links on both "near" and "far" sides of high-latency links provide a heterogeneous topology encompassing most modern wireless network characteristics.*

Table 1. Delay/Disruption Network Observations and Implications.

ID	Observation	Implication
O1	Transmitting nodes point antenna to where receiving nodes *will be* as opposed to where are.	Nodes transmit at the calculated feasible receipt time less the link transmission latency.
O2	Some links have transmission latencies larger than the communications window.	Nodes transmit during one window and receive in a subsequent window.
O3	Waiting for transmission requires the ability to store messages at local nodes.	Transmitting nodes must store messages. This may result in significant queuing delays, and therefore large queue sizes, relative to message ingest.
O4	Links are heterogeneous within the network, comprising multiple link layers and data rates. Some nodes only transmit or only receive.	Network-level protocols must handle asymmetric data rates and uni-directional links across the network.
O5	The delays and disruptions within this model neither indicate a network error nor expect a degradation of service.	Multi-phased protocols that automatically re-synchronize on a topological state change will fail to converge.

We review common networking enablers against the implications of operating within the pedagogical model.

1) *Discovery*: Node discovery relies on push or pull broadcasts to identify new network nodes. Push broadcasts come from new nodes at the time they join the network. Pull broadcasts come, periodically, from existing network nodes querying for new/unknown nodes. Frequent node mobility constantly segments the network, often into multiple partitions. The concept of network participation breaks down when the network is a set of partitions in flux. Mobile nodes would constantly broadcast themselves as they are constantly leaving and joining partitions within the network. This problem persists even when abandoning the concept of network participation and focusing more narrowly on discovering that two nodes have the ability to communicate. In such cases, mobile nodes must still constantly broadcast their capabilities hoping to catch nearby nodes. In low-latency systems, frequent broadcast simply limits the effective lifetime of the network. In high-delay networks the broadcast message may reach a receiving node after such delay that no response is possible, obviating the benefit of identifying a potential contact. In high disruption systems, the broadcast may be lost entirely.

2) *Negotiation*: Link negotiation relies on one or more round-trip communications to establish communications parameters such as data rate, session IDs, routing information. For example, the TCP/IP handshake requires three messages be exchanged between two nodes to synchronize on session information. No application data flows over TCP/IP until the handshake has completed. Higher-level protocols may impose their own negotiation once the network-layer handshakes have completed. Application data typically does not flow until all necessary protocol layers have completed their respective negotiations.

3) *State Caching*: Networks cannot be in a constant state of synchronization, as this implies they never do anything other than broadcast network state. Protocols remember communicated link states, session identifiers, routing table information, and other negotiated content in a local cache. The caching of these data implies an element of exploitable stability within the network.

4) *Bi-Directional Communications*: Protocols that establish a communication session assume that between session endpoints there is a mechanism for data exchange.

Table 2. Network function enablers are not always supported by DTNs.

Enabler\Observation	O1	O2	O3	O4	O5
Discovery	*X*	*X*			
Negotiation		*X*		*X*	
State Caching					*X*
Bi-Directional Comms				*X*	

Table 2 contains a comparison of these enablers against the network observations listed from Table 1. In this table an ***X*** represents a particular network function enabler that cannot be supported in a high-delay network based on a particular observation (and implication) of the network. Discovery cannot occur when the discovering node must consider where a discovered node will be in the future (O1).

Discovery, negotiation, and bi-direction communication cannot occur when round-trip communication incurs delays longer than the communication session itself (O2). Link negotiation cannot occur when links are significantly asymmetric, incompatible, or uni-directional. Caching the state of a network is only useful when the network state stabilizes. When state changes occur frequently and without error recovery (O5) there is no way to validate the cache.

A. Networking Enablers for Tolerant Networks

To address the problems noted above, the DTN protocols – Bundle Protocol (BP)[14], Licklider Transmission Protocol (LTP)[15], and ancillary protocols[16,17,18] – implement supplementary networking enablers that make challenged network operations more tolerant of delay and disruption. Among these key enablers are:

1) *Durable storage of data in transit*: This storage is often referred to as "store and forward" operations. Items of network traffic, termed "bundles", may be retained in storage at forwarding nodes along the end-to-end path rather than flushed from buffers within milliseconds of reception. An interruption in connectivity will merely extend the interval during which bundles are stored.

2) *Custody transfer*: End-to-end retransmission in support of network operations reliability may be impossible, within bounded time, in a challenged network: the effect of concatenating links characterized by lengthy delays and/or punctuated connectivity may be an aggregate end-to-end round trip time in excess of the useful lifetime of the data. DTN protocols instead advance the point at which bundles are retained for retransmission: as each node on the end-to-end path receives a bundle it may "take custody" of that bundle, asserting its willingness to serve as a proximate retransmission point. This strategy accelerates reliable data transmission and also mitigates the resource management problem at "upstream" nodes, which no longer need to retain copies of bundles in case of downstream data loss.

3) *Asserted communication parameters*: Transmission rates and quality of service are asserted rather than negotiated. Contact opportunities and transmission latency components are asserted rather than discovered. Asserting communication parameters prevents round-trip communication failures from impeding further transmission; the caching of negotiated and discovered network characteristics is rendered moot, since there are none.

4) *Delegation*: Recognizing that links within the network will be heterogeneous, the DTN architecture encourages delegation of environment-sensitive operational procedures to underlying layers of protocol that are tuned to the local communication environment. For example, BP runs over TCP/IP through the segment(s) of the end-to-end data path where TCP/IP operates efficiently but over LTP through the segments that are interplanetary radio links. This is especially important because retransmission timers must be computed from known round-trip communication latencies, but the round-trip latency over a deep space radio link can vary dramatically as orbital motion interrupts connectivity: a round-trip latency estimate derived from acknowledgment history, as computed by TCP, would be inaccurate in such an environment and would result in data loss and/or suboptimal bandwidth utilization.

Taken together, these additional techniques enable tolerance of all characteristics of the pedagogical model.

V. Disruption as a Special Case of Delay

Using the definitions and networking model developed in this paper, we examine the impact of delay and disruption on the services provided by the network, and the actions that protocols must take to tolerate these effects.

A. Link Layer Effects

First, we consider the communication across a single link within the network where message custody transfer has been requested. Tolerant protocols operating across this link must transmit the message and ensure that the downstream node receives and accepts custody of the message. If a custody acceptance has not been received within a certain timeout period, the message may be sent again. This scenario is illustrated in Fig. 3.

There are two significant observations regarding this situation: the propagation delays and encompassing transmission latencies are both significant and also different for the forward and return link. For link retransmission

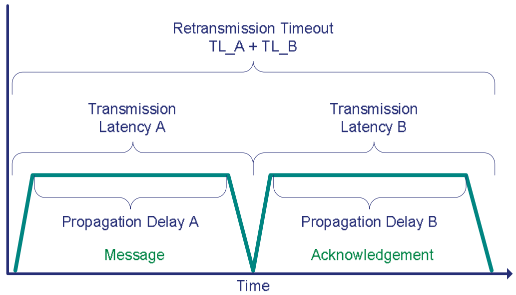

Figure 3. Latencies and delays for acknowledged message transfer. *Asymmetric links impose different transmission latencies and propagation delays.*

to succeed, delay-tolerant networks must model node mobility and delay whenever the relative velocity between nodes times the transmission latency becomes large enough to threaten the action latency over the link.

The propagation delay between mobile nodes within the network changes over time. If the initial propagation time (Delay A) is sufficiently large, the return propagation delay (Delay B) may be different because the node has moved closer or further away. Further, since the nodes are mobile while the signal is propagating, Delay A and Delay B are not static values. Rather, we assume these to be defined as the maximum expected propagation delay. The transmission delay, as previously noted, is both a function of propagation delay and data rate over a potentially asymmetric link.

This approach to delay-tolerant message transmission applies to four levels of disruption in the model, illustrated in Fig. 4.

A. Planned disruption based on mobility model, such as celestial dynamics, can be accounted for within the transmission schedule.
B. Unplanned disruption that does not occur during the transmission period has no affect on the transmission itself. As the delay increases, the probability that random, unplanned disruption in the network interferes with transmission increases. This, therefore, is a special case of well-behaved networks.
C. Unplanned disruption that occasionally conflicts with transmission is handled through the use of intelligent retransmission timers.
D. Constant disruption (link termination) or periodic, unplanned disruption that is in phase with a transmission schedule will always prevent data exchange.

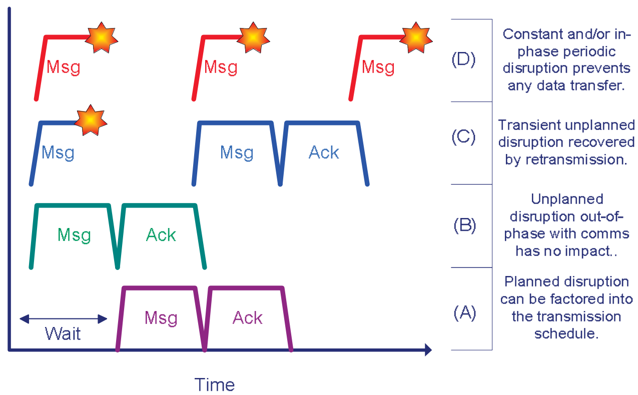

Figure 4. Effects of disruption on the delay model. *All but persistent disruption results in eventual, delayed message acknowledgement.*

At the link layer, a delay-tolerant protocol supports intelligent re-transmission timers that correct for anticipated changes in propagation delay and transmission latency. This allows the link to correct for the delays introduced by planned mobility, including disruption based on this mobility. Conversely, a disruption-tolerant protocol that does not consider delays will either attempt to negotiate an appropriate retransmission timeout (which is not possible based on our earlier discussion), or will rely on static values. When node mobility incurs a delay in the network in excess of this static parameter the retransmission mechanism will be activated even if the original message was only delayed on the link, not lost. This places multiple versions of the message on the link at once, which is inefficient. Protocols that do not support idempotent operations or otherwise trigger a flooding error-recovery phase could significantly impair the ability of the network to operate. Field tests demonstrate both the successful operation of DTN protocols in high-delay environments[19], and the ability of these protocols to avoid floods and broadcasts and non-convergence issues that hamper non-tolerant protocols such as standard IP[20].

At the link layer, we conclude that a delay-tolerant protocol handles delay, planned disruption, and unplanned disruption. A non-delay-tolerant protocol does not, and must either set timers to be excessively large (delaying retransmission) or risk redundant transmission and congestion based on inaccurate entry into a recovery mode.

B. Network Effects

Networks accomplish some data exchange function within the context of a larger system. Space networks ultimately exist to ferry science observation data to scientists. Sensor networks exist to transmit either raw or post-process results from the sensed area to downstream systems or analysts. Communication networks provide media exchange between operators. Control and monitoring data within the network exists in support of these higher-level transactions. As such, the impact of delay and disruption at the boundary of the network must be considered.

From the point of view of the network user, cases (A), (B), and (C) from Fig. 4 can easily be interpreted as disruption-less, higher-delay transmissions, stemming from the observation that propagation delays and transmission latencies in high-delay networks are asymmetric. An end user cannot distinguish between a high-delay link and a low-delay link with frequent disruption requiring frequent retransmission. As a practical matter, it is very difficult to introduce a propagation delay that breaks a human-in-the-loop use. Conversely, any disruption within the network could result in immediate data loss, placing the operator squarely in the arms of the halting problem.

Assuming that data propagation through the network is physically possible, delay-tolerant network-layer protocols seek to find the most optimal path through the network. The ability to accurately model anticipated transmission latency, and support intelligent retransmission timers, means that fewer spurious retransmissions occur within the network. Per-link disruption with fewer retransmissions provides users and operators with estimated delivery times based on delay models. Conversely, a disruption-aware protocol that does not incorporate a delay model would not incorporate delays in path prediction. Further, the additional traffic incurred by accidentally tripping into a recovery mode would reduce the overall output of the network without a clear indicator to the user or network operator as to when the network would stabilize.

As such, the practice of simulating high-delay networks (or extremely high disruption networks) by injecting periodic link disruption will likely result in protocols and associated operational concepts that do not operate as these impairments scale.

VI. Conclusion

This paper was harder to structure than anticipated. While there is an intuitive notion that non-permanent disruption models itself as delay, there is an equal counter-argument that delay can be modeled as temporary disruption. Further complicating the analysis is the fact that all networks experience some level of delay, be it a few microseconds or few petaseconds. Further, no wireless network is completely without disruption. Therefore, delay tolerance and disruption tolerance have become defined by the magnitude, not the threshold, of the phenomena.

We present an analysis of a challenged network where large delays and disruptions occur *in the absence of* error recovery. In such a network, delays and disruptions are the rule, not the exception. Protocols, therefore, that support a nominal operation and an error recovery centered around flooding and bursting activity to re-establish some state-cached baseline will find themselves with a significant, indeed endless, amount of work to do. Therefore, we re-define delay tolerance as the ability to communicate data without relying on the types of network assumptions that may not be present in high-delay networks. Namely, we claim that protocols relying on node discovery, link negotiation, state caching, and bi-directional communication are inherently not tolerant. Since some disruption-tolerant protocols do rely on these infrastructure assumptions (i.e. gaining disruption tolerance by implementing a recovery mode) they are not as scalable or general as delay-tolerant protocols. As such, we conclude that delay-tolerance is a superset of, and generally more useful than, disruption-tolerance.

In all cases, this analysis is constrained to the networking layer of the OSI model. Future work in this area would push down into the data link layer. Significantly, the flow of scheduling information between the data link and networking layers may be necessary to implement autonomous transmission schedule. Further, while work continues to quantify the performance of delay-tolerant protocols in high-delay networks, measurement is needed to quantify the performance of these protocols as the overall network delay *decreases*. This would answer the question of how costly are delay-tolerant concepts in networks experiencing only moderate or minimal delay.

Acknowledgments

This work has been funded, in part, by the National Aeronautics and Space Administration (NASA) Space Communications and Navigation (SCAN) Space DTN Readiness Project[21] and was carried out, in part, at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration..

References

[1] Cerf, V., et al. "Delay-Tolerant Networking Architecture", *RFC4838*, April 2007.

[2] Russo, R.R.; , "Wireless data communications," *Military Communications Conference, 1993. MILCOM '93. Conference record. 'Communications on the Move'., IEEE* , vol.2, no., pp.432-436 vol.2, 11-14 Oct 1993

[3] Wei Ye; Heidemann, J.; Estrin, D.; , "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *Networking, IEEE/ACM Transactions on* , vol.12, no.3, pp. 493- 506, June 2004

[4] Bianchi, G.; Borgonovo, F.; Fratta, L.; Musumeci, L.; Zorzi, M.; , "C-PRMA: a centralized packet reservation multiple access for local wireless communications," *Vehicular Technology, IEEE Transactions on* , vol.46, no.2, pp.422-436, May 1997

[5] Acharya, A.; Misra, A.; Bansal, S.; , "MACA-P: a MAC for concurrent transmissions in multi-hop wireless networks," *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on* , vol., no., pp. 505- 508, 23-26 March 2003

[6] Gang Wu; Mukumoto, K.; Fukuda, A.; , "Slotted idle signal multiple access scheme for two-way centralized wireless communication networks," *Vehicular Technology, IEEE Transactions on* , vol.43, no.2, pp.345-352, May 1994

[7] Djukic, P.; Valaee, S.; , "Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks," *Networking, IEEE/ACM Transactions on* , vol.17, no.3, pp.870-883, June 2009

[8] Irwin Jacobs, Lin-Nan Lee, Andrew Viterbi, Richard Binder, Robert Bressler, Nai-Ting Hsu, and Robert Weissler. 1977. Cpoda - a demand assignment protocol for satnet. In *Proceedings of the fifth symposium on Data communications* (SIGCOMM '77). ACM, New York, NY, USA, 2.5-2.9.

[9] Debbra Wetteroth. 2001. *OSI Reference Model for Telecommunications*. McGraw-Hill Professional.

[10] Johnson, D., Hu, Y., Maltz, D., "The Dynamic Source Routing Protocol (DSR) for Mobile As HocNetworks for IPv4", *RFC4728*, February 2007.

[11] Clausen, T., Jacquet, P., “Optimized Link State Routing Protocol (OLSR)”, RFC3626, October 2003.

[12] “Rationale, Scenarios, and Requirements for DTN in Space, Draft Informational Report”, CCSDS 734.0-G-0, December 2009.

[13] Birrane, E., Cole, R., “Network Management of Disruption-Tolerant Networks: A Systems Engineering Approach”, *AIAA SpaceOps Conference*, AIAA, 2010

[14] Scott, K., and S. Burleigh. "Bundle Protocol Specification", *RFC5050*, November 2007.

[15] Ramadas, M., Burleigh, S., Farrell, S., “Licklider Transmission Protocol Specification”, RFC5326, September 2008.

[16] Symington, S., et al., “Bundle Security Protocol Specification”, IRTF. 2010. The MITRE Corporation. Retrieved 29 Nov. 2010; http://tools.ietf.org/html/draft-irtf-dtnrg-bundle-security-17

[17] Burleigh, S., “Bundle Protocol Extended Class Of Service”, IRTF. 2010. Jet Propulsion Laboratory. Retrieved 29 Nov. 2010; http://tools.ietf.org/html/draft-irtf-dtnrg-ecos-01

[18] Symington, S., “Delay-Tolerant Networking Previous Hop Insertion Block”, IRTF. 2010. The MITRE Corporation. Retrieved 29 Nov. 2010; http://tools.ietf.org/html/draft-irtf-dtnrg-bundle-previous-hop-block-12

[19] Lindgren, Anders; Doria, Avri; , "Experiences from Deploying a Real-Life DTN System," *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE* , vol., no., pp.217-221, Jan. 2007

[20] Brenton Walker, Masato Tsuru, Armando Caro, Ari Keranen, Jorg Ott, Teemu Karkkainen, Shinya Yamamura, and Akira Nagata. 2010. The state of DTN evaluation. In *Proceedings of the 5th ACM workshop on Challenged networks* (CHANTS '10). ACM, New York, NY, USA, 29-30. DOI=10.1145/1859934.1859941 http://doi.acm.org/10.1145/1859934.1859941

[21] “DTN Goals, Demonstration Phasing and Participants”, Space Communications and Navigation Homepage. 2008. NASA. Retrieved 29 Nov. 2010; https://www.spacecomm.nasa.gov/spacecomm/programs/technology/dtn/goals.cfm